

50325-0641 (Seq. No. 4831)

*Patent*

UNITED STATES PATENT APPLICATION

FOR

METHOD AND APPARATUS FOR DEPLOYING CONFIGURATION INSTRUCTIONS TO  
SECURITY DEVICES IN ORDER TO IMPLEMENT A SECURITY POLICY ON A NETWORK

INVENTORS:

SHIGANG CHEN  
PARTHA BHATTACHARYA

PREPARED BY:

HICKMAN PALERMO TRUONG & BECKER LLP  
1600 WILLOW STREET  
SAN JOSE, CA 95125  
(408) 414-1080

EXPRESS MAIL MAILING INFORMATION

"Express Mail" mailing label number EL734971445US

Date of Deposit February 15, 2002

METHOD AND APPARATUS FOR DEPLOYING CONFIGURATION INSTRUCTIONS TO  
SECURITY DEVICES IN ORDER TO IMPLEMENT A SECURITY POLICY ON A  
NETWORK

FIELD OF THE INVENTION

[0011] The present invention generally relates to configuring security devices on a network. The invention relates more specifically to a method and apparatus for deploying configuration instructions to security devices in order to implement a security policy on a network.

BACKGROUND OF THE INVENTION

[0012] Network management systems implement policies by configuring managed devices in the network, such as routers, switches, gateways, and firewalls, with instructions appropriate to carry out a desired policy. One general application for a network management system is implementation of a security policy on firewalls and other security devices. A policy server may be used in conjunction with other components to configure security devices on the network with security policies. CISCO SECURE POLICY MANAGER (CSPM), manufactured by CISCO SYSTEMS, INC., San Jose, California, is a commercially available product for implementing security policies using a policy server.

[0013] Network management systems are increasingly required to manage larger networks that have numerous firewalls, virtual private networks, and other security devices. To implement security policies properly, an administrator must configure all such devices properly. Consequently, security management can become a difficult task on large networks. A centralized management scheme is often employed for large networks because such systems are efficient, and retain all security information and configurations in one location.

In addition, centralized management schemes reduce the possibility of inconsistent or incompatible configurations for different security devices.

[0014] With centralized management schemes, security policy configurations are typically deployed from a management site to many security devices using telnet or file transfer protocol, in clear packet traffic or via encrypted tunnel (e.g. IPSec). The use of centralized management stations in this manner requires firewalls and other security devices to always be open. This use of a centralized management station is typically referred to as a permit management property.

[0015] Often, networks use a string of security devices in series. The management host maintains all channels between the security devices open for any new set of configurations that it generates.

[0016] When address translation mechanisms are used in centralized management schemes, the order in which security devices become configured determines whether deployment of a set of configurations to multiple security devices will be successfully implemented. Even if the new configurations for the security policy satisfy the permit management property, some management channels may be blocked in the middle of deployment because some security devices have been configured while others have not. When a mixture of new and old configurations result, the management channels that are blocked during the transition can prevent some firewalls from receiving their new configurations. Consequently, the entire network goes into a partially configured state with unpredictable security behavior.

[0017] Furthermore, because some management channels are blocked, it becomes more difficult to roll-back security devices to their original state. This can lead to security leaks or service disruptions for an extended period of time.

[0018] Based on the foregoing, there is a need for determining the order in which a new set of configurations for security devices on a network can be deployed so that newly deployed configurations do not block other configurations that are being deployed when implementing a new security policy.

[0019] The approaches described in this section could be pursued, but are not necessarily approaches that have been previously conceived or pursued. Therefore, unless otherwise indicated herein, the approaches described in this section are not prior art to the claims in this application and are not admitted to be prior art by inclusion in this section.

## SUMMARY OF THE INVENTION

[0010] The foregoing needs, and other needs and objects that will become apparent for the following description, are achieved in the present invention, which comprises, in some aspects, a method, system and device that deploys configuration instructions to a plurality of security devices for the purpose of implementing a security policy on a network.

[0011] According to one embodiment, implementation of a security policy is detected as causing an address translation alteration between a management source and a plurality of security devices. One or more sets of security devices are identified that each have one or more configuration dependencies as a result of the address translation alteration. Configuration instructions are sent from the management source to each of the one or more sets of security devices using an order determined by the identified configuration dependencies. The configuration instructions are used to implement the security policy on the network.

## BRIEF DESCRIPTION OF THE DRAWINGS

[0012] The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[0013] FIG. 1 is a flow diagram that illustrates a method for deploying configuration instructions to security devices on a network.

[0014] FIG. 2 is a flow diagram that illustrates a method for identifying a sequence for deploying configuration instructions to a set of security devices on a network path containing an address translation.

[0015] FIG. 3 is a block diagram that illustrates a system for configuring security devices on a network having a new natural address translation component.

[0016] FIG. 4 is a block diagram that illustrates a system for configuring security devices on a network having a new reverse static natural address translation component.

[0017] FIG. 5 is a block diagram that illustrates a system for configuring security devices on a network having a new tunneling translation component.

[0018] FIG. 6 is a block diagram that illustrates an embodiment for deploying configuration instructions to multiple sets of dependent security devices in parallel.

[0019] FIG. 7 is a block diagram that illustrates a computer system upon which an embodiment may be implemented.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

**[0020]** A method and apparatus for deploying configuration instructions to a plurality of security devices in order to implement a security policy on a network is described. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

**[0021]** Embodiments are described herein according to the following outline:

- 1.0 OVERVIEW
- 2.0 METHODOLOGY FOR DEPLOYING CONFIGURATION INSTRUCTIONS TO SECURITY DEVICES
- 3.0 ADDRESS TRANSLATIONS ON NETWORK PATHS
- 4.0 PARALLEL DEPLOYMENT OF CONFIGURATION INSTRUCTIONS
- 5.0 IMPLEMENTATION MECHANISMS-HARDWARE OVERVIEW
- 6.0 EXTENSIONS AND ALTERNATIVES

### 1.0 OVERVIEW

**[0022]** According to one embodiment, implementation of a security policy is detected in the form of one or more operations causing an address translation alteration in data packets that are communicated between a management source and one or more security devices. One or more sets of security devices are identified that would each have one or more configuration dependencies as a result of the address translation alteration, if the security policy is implemented. Configuration instructions are sent from the management source to each of the one or more sets of security devices using an order that is determined based on

the identified configuration dependencies. The configuration instructions are used to implement the security policy on the network.

[0023] As used herein, a “management source” is a policy server, or other computer readable-medium capable of sending instructions to the security devices. In one example, a workstation may access a repository to send out configuration instructions to the security devices over the network. A “security device” refers to any module, device or component of a network that screens or regulates packet traffic.

[0024] The term “configuration dependency” means that the configuration of one security device is dependent on another security device being configured or not configured. By way of example, a configuration dependency corresponds to a first security device having to receive its configuration instructions before a second security device is configured, else the second security device would block the configuration instructions to the first security device.

[0025] In one aspect, configuration instructions are sent from the management source to each of the one or more sets of security devices in parallel, for purpose of implementing the security policy on the network. As used herein, the term “in parallel” means at one time or nearly simultaneously. Two security devices that are configured in parallel means that the two security devices were configured independent of one another. The term “set” refers to one or more.

[0026] Using the approaches disclosed herein, a new set of configuration instructions may be implemented for security devices on a network without risk that some newly configured security devices will block other security devices from receiving their respective configuration instructions. Further, a new set of configuration instructions may be deployed according to a schedule. The schedule may permit one group of security devices to be configured in parallel with another group of security devices, based on a determination that



the one group of security devices is not dependent on the other group of security devices when receiving configuration instructions from a management host. Use of the schedule enables multiple security devices to be configured in parallel across large networks, thereby reducing the time it takes to implement a new security policy on that network.

## 2.0 METHODOLOGY FOR DEPLOYING CONFIGURATION INSTRUCTIONS TO SECURITY DEVICES

[0027] FIG. 1 illustrates a method for deploying configuration instructions to security devices on a network. The configuration instructions may be used to configure the security devices for the purpose of implementing or updating a security policy.

[0028] In step 110, a new security policy is identified for the security devices that are in a network. The new security policy may require the security devices of the network to be individually configured with configuration instructions sent from a management source. Block 110 may involve an administrator determining that a new security policy is needed in the network, or it may involve a software element determining that a new security policy needs to be deployed to the network, automatically or based on user input.

[0029] In step 120, a determination is made as to whether the new security policy creates a change in configuration dependencies on the network, as compared with an old or existing security policy on the network. Each configuration dependency may correspond to a security device that must receive its configuration instructions before another security device is configured. As will be discussed, configuration dependencies may be created as a result of utilizing a device or other mechanism that performs an address translation on the network. For example, the new security policy may require that a network address translator be

positioned between the management source and one of a series of security devices forming a network path.

**[0030]** If the determination in step 120 is negative, then in step 130, security devices on the network are configured to implement the new security policy without consideration for configuration dependencies. No new sequence for deploying configuration instructions to the security devices is necessary. Step 130 may be carried out, for example, by issuing appropriate configuration commands using SNMP, or a similar protocol, from a management station that is executing a network security policy management application. An example of such an application is CiscoSecure Policy Manager, from Cisco Systems, Inc., San Jose, California.

**[0031]** If the determination in step 120 is positive, then in step 140, one or more configuration dependencies, which would be created as a result of implementing the new security policy, are identified. Sets of security devices that form network paths containing the one or more configuration dependencies may be identified. Information identifying the sets may be stored in memory.

**[0032]** In step 150, the identified configuration dependencies are used to determine a deployment order for sending configuration instructions to individual security devices on the network when implementing the new security policy. The deployment order ensures that security devices that are part of the one or more configuration dependencies are configured in a successful sequence. The deployment order may also specify that one or more security devices are to be configured at the same time, so as to configure multiple security devices in parallel. The deployment order may comprise an ordered list of devices or sets of devices as determined in step 140.

[0033] In step 160, configuration instructions for implementing the new security policy are deployed to individual security devices, according to the deployment order determined in step 150. Deployment may occur to individual devices or to the sets of devices that were determined in step 140.

[0034] FIG. 2 illustrates a method for identifying a deployment order for sending configuration instructions to a set of security devices on a network path containing an address translation. FIG. 2 represents a more detailed view of sub-steps that can be carried out as part of steps 120, 140, 150 of FIG. 1.

[0035] In step 210, a determination is made that implementation of a new security policy will cause an address translation alteration on packets that flow through the devices. The presence of a device performing an address translation in between the management source and the sequence of security devices receiving the configuration instructions may create configuration dependencies. As a result, the configuration instructions may have to be deployed in a designated order so that each of the security devices can receive and process those configuration instructions. Specific examples of address translations and resulting configuration dependencies is described and illustrated with FIG. 3, FIG. 4 and FIG. 5.

[0036] In an embodiment, an address translation corresponds to a device that performs a (i) a static network address translation (NAT), (ii) a hiding NAT, (iii) a reverse static NAT, or (iv) a tunneling translation. For each type of address translation, an address translation alteration corresponds to a device that (i) adds an address translation with a new security policy when no address translation existed with the old security policy, (ii) deletes an address translation that existed with the old security policy, and (iii) implements a new address translation with a new security policy when the old security policy had a different address translation.

[0037] In an embodiment, an address translation alteration may be performed by a device, such as a firewall, that implements one or more rules that correspond to the address translation alterations. For example, the addition of an address translation may be the result of configuring a firewall to execute a new rule corresponding to that address translation. Similarly, the deletion of an address translation may be the result of configuring a firewall to delete, modify or supplement an existing rule that was executed to perform the address translation.

[0038] In step 220, one or more network paths containing configuration dependencies are identified. In an embodiment, step 220 involves identifying paths having an origin at the device where the address translation alteration takes place. For any one identified network path, the ability of one or more security devices, referred to as “dependent security devices”, to receive configuration instructions will depend on whether other security devices on those same network paths are configured or not configured. Each dependent security device is part of a configuration dependency that results from the address translation alteration.

[0039] For example, a network path may contain a set of security devices aligned in series, designated  $FW_1 \dots FW_n$ , where  $FW_1$  is the security device that is adjacent to the address translation alteration. In this scenario, other security devices downstream from  $FW_1$  may be dependent on  $FW_1$ . In some cases, other security devices downstream from  $FW_1$  may be dependent on  $FW_1$  and intermediate security devices.

[0040] In step 230, security devices in each network path are configured in a reverse (or last-first) order or sequence. In a reverse order, the last security device on the network path is configured first, sequentially followed by other security devices on the network path, until the security device adjacent to the address translation alteration is configured. The last security device corresponds to the last security device in the series to receive its

configuration instructions from the management source. In the example provided, the deployment order for the set of security devices on that network path would be:  $FW_n \dots FW_1$ , with the left position being the starting point.

[0041] Other embodiments may use other orders or sequences to deploy configuration instructions to dependent security devices. In addition, the devices that form the address translation alteration, and which are between the management source and the address translation alteration, may also be part of the determined order. For example, in the tunneling translation described with FIG. 5, the two firewalls used to form the tunneling translation are configured in a last-first order, relative to the management source.

### 3.0 ADDRESS TRANSLATIONS ON NETWORK PATHS

[0042] FIG. 3 is a block diagram that illustrates a system for configuring security devices on a network having a new natural address translation component. FIG. 4 is a block diagram that illustrates a system for configuring security devices on a network having a new reverse static natural address translation component. FIG. 5 is a block diagram that illustrates a system for configuring security devices on a network having a new tunneling translation component. FIG. 6 is a block diagram that illustrates an embodiment for deploying configuration instructions to multiple sets of dependent security devices in parallel. Thus, FIGS. 3-6 illustrate networks in which a management host implements a new security policy on a plurality of firewalls, where the new security policy also implements an intervening address translation between the management host and one or more of the plurality of security devices.

[0043] Each management host described in FIGS. 3-6 may be assumed to be part of a centralized management scheme. Furthermore, each management host described in FIGS. 3-6

may be assumed to deploy configuration instructions using a permit management property. For purpose of explanation, each address translation alteration discussed in FIGS. 3-6 is also assumed to be an addition. Embodiments of the invention are also intended to include modifications and deletions of address translations from a network path.

**[0044]** FIG. 3 illustrates a system for configuring security devices on a network having a new network address translation (NAT) component. In FIG. 3, a management host 310 is connected on network 300 to a first firewall 330, second firewall 340, and third firewall 350. The first firewall 330, second firewall 340 and third firewall 350 are positioned in series, such that first firewall 330 is the closest to the management host 310, and third firewall 350 is furthest from the management host 310.

**[0045]** A NAT component 320 is positioned to intercept communications sent from management host 310 to first firewall 330. A first channel 322 connects the management host 310 to the NAT component 320, so that the NAT component and management host are logically adjacent to one another. The NAT component 320 may be a configuration, software element or module of first firewall 330. For example, one or more rules may be executed by the first firewall in order to carry out the network address translation functions of NAT 320. A second channel 332 connects the first firewall 330 to a second firewall 340, so that the first and second firewalls are logically adjacent to one another. A third channel 342 connects a third firewall 350 to the second firewall 340. The NAT 320, first firewall 330, second firewall 340 and third firewall 350 are arranged in a logical network path 355.

**[0046]** Each data packet received by NAT component 320 includes a source address identifier that is altered by the NAT component. The NAT component 320 converts the source address identifier into a new source address identifier for components that are downstream from the NAT along network path 355.

**[0047]** In a typical scenario, the NAT component 320 is used to translate a limited set of exterior Internet Protocol (IP) addresses into a greater set of interior or local IP addresses. Two configurations are possible for using NAT component 320. The NAT component 320 may be utilized for either static or hiding type network address translation. In static network address translation, one interior local address is mapped to a corresponding (IP) address, or other exterior address. In hiding network address translation, multiple interior network addresses are mapped to one or more exterior IP addresses, or other exterior addresses. There are many instances in which NAT component 320 may be employed. For example, NAT component 320 may be used to enable an Internet Service Provider to translates a limited number of exterior addresses into a large number of interior addresses, thereby giving many users access to the Internet while using only a limited supply of IP addresses. As another component, NAT component 320 may be used to hide identification information from traffic leaving the local portion of network 300. Further, hiding type NAT enables a local network to hide the actual addresses of local network devices from external network elements, while enabling packets to be routed to the local network devices.

**[0048]** The management host 310 can send configuration instructions to each of the first firewall 330, second firewall 340, and third firewall 350 in order to implement a security policy on the network 300. The channels 322, 332 and 342 link first firewall 330, second firewall 340 and third firewall 350, respectively, to management host 310 in series. As such, configuration instructions sent from management host 310 to third firewall 350 must pass through first firewall 330 and second firewall 340.

**[0049]** The addition of NAT component 320 causes a new configuration dependency to form on network path 355. Configuration dependencies may cause deployment order problems when a new security policy is to be implemented on network 300. For example,

NAT component 320 may be added as part of a new security policy that includes re-configuration of first firewall 330, second firewall 340, and third firewall 350. If the deployment order is ignored, then once NAT component 320 is added to network path 355, new configuration instructions sent from management host 310 to first firewall 330, second firewall 340 and/or third firewall 350 may be dropped rather than implemented. This can open the network up to a security breach, and under certain circumstances, cause network 300 to shut down.

**[0050]** The approaches herein address this possibility by enabling management host 310 to deploy configuration instructions for implementing a new security policy using a schedule, so that deployment order problems are avoided when NAT component 320 is added to network path 355. Approaches herein ensure correct deployment orders for when management host 310 sends configuration instructions to the firewalls on network path 355. Furthermore, on larger networks with multiple branches stemming from or passing through NAT component 320, configuration instructions may be sent to non-dependent firewalls simultaneously, or otherwise in parallel.

**[0051]** The addition of NAT 320 to network path 355 illustrates one type of configuration dependency that may arise. As an example, assume that addition of NAT component 320 causes data packets that contain internal address identifiers for a component X to identify a component Y instead. Under an existing configuration, the first firewall 330, second firewall 340 and third firewall 350 may be configured to receive data packets containing source address identifier X, but the new security policy will require that each of those firewalls receive data packets having a source identifier of Y, not X. If NAT component 320 is added to network path 355, but second firewall 340 and third firewall 350 are not configured, then the second and third firewalls are configured to receive data packets containing the source



identifier X; however, the presence of NAT component 320 causes configuration instructions (and other traffic) sent from management host 310 to second firewall 340 or third firewall 350 to have source address identification component Y. Consequently, the configuration instructions sent to second and third firewalls 340 and 350 would be dropped, since those firewalls would be able to receive only data packets having the source address identifier X.

[0052] Similarly, if second firewall 340 is configured before third firewall 350, then the second firewall would drop all traffic, including configuration instructions, that passes through second firewall 340 directed to third firewall 350. Third firewall 350 requires such traffic to have X as its source address identifier component, but second firewall 340 would let only traffic having Y as its source address identifier component to be passed-through.

[0053] Accordingly, a first configuration dependency that exists on network path 355 from the addition of NAT component 320 is that third firewall 350 must receive configuration instructions that are forwarded from first firewall 330 and second firewall 340. Third firewall 350 would not be configurable if, for example, second firewall 340 was to drop its configuration instructions. A second configuration dependency that exists on network path 355 from the addition of NAT component 320 is that second firewall 340 must receive configuration instructions that are forwarded from first firewall 330. Therefore, second firewall 340 would not be configurable if first firewall 330 was to drop its configuration instructions from management device 310.

[0054] The approaches herein enable deploying configuration instructions from management host 310 to each firewall on network path 355 without dropping any of the instructions as a result of configuration dependencies on that network path. Specifically, embodiments determine an order for deploying configuration instructions that does not cause

any device to drop configuration instructions as a consequence of the configuration dependencies.

[0055] In one embodiment shown by FIG. 1, configuration instructions  $CI(FW_3)$  sent from management host 310 to third firewall 350 are sent first, at time  $t=1$ . At time  $t=2$ , configuration instructions  $CI(FW_2)$  are sent to second firewall 340. Then at time  $t=3$ , configuration instructions  $CI(FW_1)$  are sent to first firewall 330. The configuration instructions  $CI(FW_1)$  may cause first firewall 330 to implement that NAT component 320. The result is that a last-first sequential deployment order is used to configure the series of firewalls on network path 355 using configuration instructions sent from management host 310.

[0056] In an embodiment, firewalls and other security devices on network path 355 may be configured using a schedule that controls the order in which configuration instructions are deployed. The schedule may specify that  $CI(FW_3)$  is to be deployed at  $t=1$ ,  $CI(FW_2)$  is to be deployed at  $t=2$ , and  $CI(FW_1)$  is to be deployed at  $t=3$ . Moreover, the schedule may provide for security devices on other network paths (not shown) to be deployed in parallel. For example, management host 310 may deploy  $CI(FW_3)$  at the same time as  $CI(FW_n)$  on another network path (not shown) that stems from NAT component 320. Sending configuration instructions in parallel to firewalls and other security devices that are not dependent on one another saves time when a new security policy is to be implemented on the network.

[0057] FIG. 4 illustrates a system for configuring security devices on a network having a new, reverse static network address translation (NAT) component. In FIG. 4, a management host 410 is connected on network 400 to a first firewall 420, second firewall 430, and third firewall 450. The first firewall 420, second firewall 430 and third firewall 450 are positioned in series, such that the first firewall 420 is the closest to the management host 410, and the

third firewall 450 is furthest from the management host 410. The second firewall 440 has a static NAT component 440 as a configuration, software module or other integrated component. The static NAT component 440 carries out address translation on packets communicated between second firewall 430 and third firewall 450. The static NAT component 440 is illustrated in a position that is reversed with respect to the position of NAT component 320 in relation to firewall 330 of FIG. 3, because management host 410 is external, and is seeking access to the internal components of a local network. The security devices on network path 455 screen incoming communications for a particular destination address identifier.

**[0058]** A first channel 422 connects the management host 410 to the first firewall 420. A second channel 432 connects the first firewall 420 to the second firewall 430 and static NAT component 440. A third channel 442 connects the second firewall 430 and static NAT component 440 to the third firewall 450, so as to form a logical network path 455.

**[0059]** The static NAT component 440 serves to convert address identification information in packets received from management host 410. In an example shown, destination address identification component X may be replaced with Y by static NAT component 440. In the presence of static NAT component 440, a destination address component of traffic directed to FW<sub>3</sub> is Y. In the absence of static NAT component 440, the destination address component of traffic to FW<sub>3</sub> is X.

**[0060]** A configuration such as that shown with FIG. 4 may be used when second firewall 430 and static NAT component 440 mark a boundary of a local network containing third firewall 450. Management host 410 may be external to the local network. The static NAT component 440 is used to hide the addresses of internal components of the local network to external devices, such as management host 410. The reverse static NAT component 440

precludes management host 410 from communicating with third firewall 450 using that firewall's actual address.

[0061] Certain configuration dependencies form on network path 455 after second firewall 430 is configured to include static NAT 440. As will be described, an example of a first dependency is that third firewall 450 must be configured before or after both first firewall 420 and second firewall 430.

[0062] For example, assume that management host 410 sends configuration instructions to first firewall 420, before configuring third firewall 450 and second firewall 430 to include static NAT 440. Then, a subsequent attempt to configure third firewall 450 would fail. First firewall 420 would expect the destination address component of third firewall to be Y, but third firewall 450 would expect its control instructions to specify a destination address of X. As a result, the configuration instructions to third firewall 450 would be dropped at either first firewall 420 or third firewall 450. A similar result would occur if the second firewall 420 is configured before third firewall 450, but first firewall 420 is configured after the third firewall.

[0063] Accordingly, network path 455 may be configured in one of two ways. First firewall 420 and second firewall 430 may be configured before third firewall 450, with configuration instructions sent from management host 410 for third firewall 450 identifying Y as a destination address component for the third firewall. Alternatively, third firewall 450 may be configured before first firewall 420 and second firewall 430, with configuration instructions sent from management host 410 for the third firewall identifying X as a destination address component for the third firewall.

[0064] A schedule to deploy configuration instructions may be used to efficiently configure the firewalls on network 455 while taking into account the configuration

dependencies. As with other embodiments described, a last-first methodology may be employed to configure the firewalls on network path 455. At  $t=1$ , configuration instructions  $CI(FW_3)$  416 are sent from management host 410 to third firewall 450. At  $t=2$ , configuration instructions  $CI(FW_2)$  414 are sent from management host 410 to second firewall 430. At  $t=2$  or at  $t=3$ , configuration instructions  $CI(FW_1)$  412 are sent to first firewall 420.

[0065] FIG. 5 illustrates a system for configuring security devices on a network having a tunneling translation component. In FIG. 5, a management host 510 is connected on network 500 to a first firewall 520, second firewall 530, and third firewall 540. The first firewall 520, second firewall 530 and third firewall 540 are positioned in series, with the first firewall 520 being the closest to the management host 510, and the third firewall 540 being the furthest from the management host 510. A first channel 522 connects the management host 510 to the first firewall 520. A tunnel 532 is established between the first firewall 520 and the second firewall 530. A second channel connects the second firewall 530 to the third firewall 540. The first channel 522, second channel 542, and tunnel 532 serve to connect the first firewall 520, second firewall 530 and third firewall 540 in series, so as to form a network path 555.

[0066] The tunnel 532 encrypts, authenticates and performs other security measures for data packets passing from first firewall 520 to second firewall 530. In addition, tunnel 532 translates both source and destination address components of traffic passing through the tunnel. Since this traffic would include configuration instructions passing from management component 510 to third firewall 540, configuration dependencies are formed on network path 555 when new security policies are implemented.

[0067] One configuration dependency in the presence of tunnel 532 is that firewalls on one side of the tunnel should all be configured before firewalls on another side of the tunnel. For example, in FIG. 3, first firewall 520 and second firewall 530 should both be configured

before the third firewall 540. Alternatively, the third firewall 540 should be configured before both the first firewall 520 and the second firewall 530. If the first firewall 520 is configured before the third firewall 540, and the second firewall 530 is not yet configured when third firewall component 540 is to be configured, then first firewall 520 would drive the configuration instructions for the third firewall into the tunnel 532, but second firewall 530 would not be able to retrieve those configuration instructions. The same result would occur if second firewall 530 is configured before first firewall 520 is configured, and an attempt to configure third firewall 540 occurred.

[0068] Accordingly, in the approaches herein, when a tunnel is connecting two firewalls on a network path, both of the two firewalls are configured either before or after the other firewalls on the network path. In FIG. 5, first firewall 520 and second firewall 530 are configured either before or after third firewall 540.

[0069] As with other embodiments, a last-first deployment order is performed when tunnel 532 is employed on network path 555. A schedule may provide that at  $t=1$ , configuration instructions  $CI(FW_3)$  516 are sent to third firewall 540. Next, at  $t=2$ , configuration instructions  $CI(FW_2)$  514 are sent to second firewall 530. At  $t=2$  or 3, configuration instructions  $CI(FW_1)$  512 are sent to first firewall 520. Thus, some parallelism may be employed in that  $CI(FW_2)$  514 and  $CI(FW_1)$  512 may be sent at the same time from management host 510 to the respective firewalls.

#### 4.0 PARALLEL DEPLOYMENT OF CONFIGURATION INSTRUCTIONS

[0070] FIG. 6 illustrates an embodiment for deploying configuration instructions to multiple sets of dependent security devices in parallel. In an embodiment, each dependent set of security devices refers to a network path having at least one configuration dependency,

where one security device can only receive configuration instructions if another security device is configured or not configured. For purpose of explanation, a single natural address translation is described.

[0071] In FIG. 6, a network 600 includes a management host 610 connected to a first firewall 620 via a channel 612. A NAT component 630 is configured with first firewall 620. A first channel 622 connects first firewall 620 to management host 610.

[0072] In the example provided, a first network path 645, second network path 655, and third network path 665 each stem from NAT component 630. Each of the network paths includes at least one firewall adjacent to NAT component 630. First network path 645 includes firewall 640 (FW<sub>11</sub>) connected to NAT component 630 via channel 632. Second network path 655 includes firewall 650 (FW<sub>21</sub>) connected to NAT component via channel 634. Third network path 665 includes firewall 660 (FW<sub>m1</sub>) that connects to NAT component 630 via third channel 636.

[0073] Each network path 645, 655 and 665 includes a last firewall. For example, first network path 645 includes last firewall 646 (FW<sub>1n</sub>), second network path 655 includes last firewall 656 (FW<sub>2n</sub>), and third network path 665 includes last firewall 666 (FW<sub>mn</sub>). The last firewall corresponds to the last one of the firewalls on the respective firewalls to receive configuration instructions from the management host in series.

[0074] When the address translation alteration is caused by the addition of NAT component 630, configuration dependencies are created downstream from the NAT component, along the first, second and third network paths. For example, for each network path, the last firewall is dependent on the upstream firewall adjacent to NAT component 630 for its configuration instructions.

[0075] In one embodiment, configuration instructions are deployed to the firewalls on the first, second and third network paths 645, 655, 665, using a last-first order. On first network path 645, last firewall 646 ( $FW_{1n}$ ) receives its configuration instructions before the firewall 640 ( $FW_{11}$ ). On second network path 655, last firewall 656 ( $FW_{2n}$ ) receives its configuration instructions before firewall 650 ( $FW_{21}$ ). And on third network path 665, last firewall 666 ( $FW_{mn}$ ) receives its configuration instructions before firewall 660 ( $FW_{m1}$ ). Intermediate firewalls between the NAT component 630 and the last firewalls 646, 656 and 666 on respective network paths 645, 655 and 665 also receive configuration instructions in the last-first order.

[0076] Each network path 645, 655, 665 is independent of the other network paths for receiving configuration instructions from management host. This is because NAT component 630 communicates with adjacent firewalls of the respective network paths directly. As a result, configuration instructions to each network path 645, 655 and 665 may be deployed in parallel.

[0077] In one embodiment, configuration instructions 614 that are issued to the last firewalls 646, 656, 666 are sent together, at  $t=0$ . Still further, an embodiment provides that at another time interval, configuration instructions for the next-to-last firewall on each respective network path is deployed. If each network path 645, 655, 665 is assumed to have the same number of firewalls, then configuration instructions 621 to the respective adjacent firewalls 640, 650, 660 are deployed together, at some time  $t=i$ .

[0078] FIG. 6 illustrates one example of parallel deployment of configuration instructions to firewalls on network paths. Other examples may be provided. For example, in FIG. 5, if another firewall is added in series with third firewall 540, then that configuration instructions to that firewall, and to the third firewall 540 may be deployed in parallel. Therefore, in some



scenarios, configuration instructions may be deployed to firewalls aligned in series on the same network path.

## 5.0 IMPLEMENTATION MECHANISMS -- HARDWARE OVERVIEW

[0079] FIG. 7 is a block diagram that illustrates a computer system 700 upon which an embodiment of the invention may be implemented. Computer system 700 includes a bus 702 or other communication mechanism for communicating information, and a processor 704 coupled with bus 702 for processing information. Computer system 700 also includes a main memory 706, such as a random access memory ("RAM") or other dynamic storage device, coupled to bus 702 for storing information and instructions to be executed by processor 704. Main memory 706 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 704. Computer system 700 further includes a read only memory ("ROM") 708 or other static storage device coupled to bus 702 for storing static information and instructions for processor 704. A storage device 710, such as a magnetic disk or optical disk, is provided and coupled to bus 702 for storing information and instructions.

[0080] Computer system 700 may be coupled via bus 702 to a display 712, such as a cathode ray tube ("CRT"), for displaying information to a computer user. An input device 714, including alphanumeric and other keys, is coupled to bus 702 for communicating information and command selections to processor 704. Another type of user input device is cursor control 716, such as a mouse, trackball, stylus, or cursor direction keys for communicating direction information and command selections to processor 704 and for controlling cursor movement on display 712. This input device typically has two degrees of

freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

[0081] The invention is related to the use of computer system 700 for deploying configuration instructions to a plurality of security devices in order to implement a security policy on a network is described. According to one embodiment of the invention, deploying configuration instructions to a plurality of security devices in order to implement a security policy on a network is described is provided by computer system 700 in response to processor 704 executing one or more sequences of one or more instructions contained in main memory 706. Such instructions may be read into main memory 706 from another computer-readable medium, such as storage device 710. Execution of the sequences of instructions contained in main memory 706 causes processor 704 to perform the process steps described herein. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and software.

[0082] The term "computer-readable medium" as used herein refers to any medium that participates in providing instructions to processor 704 for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 710. Volatile media includes dynamic memory, such as main memory 706. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 702. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0083] Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punchcards, papertape, any other physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

[0084] Various forms of computer readable media may be involved in carrying one or more sequences of one or more instructions to processor 704 for execution. For example, the instructions may initially be carried on a magnetic disk of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system 700 can receive the data on the telephone line and use an infra-red transmitter to convert the data to an infra-red signal. An infra-red detector can receive the data carried in the infra-red signal and appropriate circuitry can place the data on bus 702. Bus 702 carries the data to main memory 706, from which processor 704 retrieves and executes the instructions. The instructions received by main memory 706 may optionally be stored on storage device 710 either before or after execution by processor 704.

[0085] Computer system 700 also includes a communication interface 718 coupled to bus 702. Communication interface 718 provides a two-way data communication coupling to a network link 720 that is connected to a local network 722. For example, communication interface 718 may be an integrated services digital network ("ISDN") card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 718 may be a local area network ("LAN") card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 718 sends and receives

electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[0086] Network link 720 typically provides data communication through one or more networks to other data devices. For example, network link 720 may provide a connection through local network 722 to a host computer 724 or to data equipment operated by an Internet Service Provider ("ISP") 726. ISP 726 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the "Internet" 728. Local network 722 and Internet 728 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 720 and through communication interface 718, which carry the digital data to and from computer system 700, are exemplary forms of carrier waves transporting the information.

[0087] Computer system 700 can send messages and receive data, including program code, through the network(s), network link 720 and communication interface 718. In the Internet example, a server 730 might transmit a requested code for an application program through Internet 728, ISP 726, local network 722 and communication interface 718. In accordance with the invention, one such downloaded application provides for deploying configuration instructions to a plurality of security devices in order to implement a security policy on a network is described, as described herein.

[0088] The received code may be executed by processor 704 as it is received, and/or stored in storage device 710, or other non-volatile storage for later execution. In this manner, computer system 700 may obtain application code in the form of a carrier wave.

## 6.0 EXTENSIONS AND ALTERNATIVES

[0089] In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

---